

Final Exam

CMSY-199, Spring 2014

Circle the letter of the best response for each item.

1. What class is the root of the Java class hierarchy?
 - (a) `java.awt.Object`
 - (b) `java.io.Object`
 - (c) `java.lang.Object`
 - (d) `java.util.Object`
2. Which of the following represents the *is a* relationship between classes?
 - (a) Inheritance
 - (b) Composition
 - (c) Dependency
 - (d) Realization
3. Which of the following represents the *has a* relationship between classes?
 - (a) Inheritance
 - (b) Composition
 - (c) Dependency
 - (d) Realization
4. The mechanism by which a superclass variable invokes an overridden method in a subclass is called
 - (a) Abstraction
 - (b) Encapsulation
 - (c) Information Hiding
 - (d) Polymorphism
5. Which of the following may *not* contain any method implementations?
 - (a) abstract class
 - (b) interface
 - (c) subclass
 - (d) superclass

6. Which of the following is a runtime exception thrown by the JVM?
- (a) `ClassCastException`
 - (b) `NullPointerException`
 - (c) `ArrayIndexOutOfBoundsException`
 - (d) All of the above
7. The size (in bits) of the primitive `float` type in Java is
- (a) 8
 - (b) 16
 - (c) 32
 - (d) 64
8. What character encoding set does Java use to represent characters?
- (a) EBCDIC
 - (b) Kanji
 - (c) Unicode
 - (d) XML
9. A car dealership needs a program to store information about the cars for sale. For each car, they want to keep track of the following information: number of doors (2 or 4), whether the car has air conditioning, and its average number of miles per gallon. Which of the following is the best design?
- (a) Use one class, `Car`, which has three data fields: `int numDoors`, `boolean hasAir`, and `double milesPerGallon`.
 - (b) Use four unrelated classes: `Car`, `Doors`, `AirConditioning`, and `MilesPerGallon`.
 - (c) Use a class `Car` which has three subclasses: `Doors`, `AirConditioning`, and `MilesPerGallon`.
 - (d) Use a class `Car`, which has a subclass `Doors`, with a subclass `AirConditioning`, with a subclass `MilesPerGallon`.
 - (e) Use three classes: `Doors`, `AirConditioning`, and `MilesPerGallon`, each with subclass `Car`.

10. Consider the following Java classes:

```
public class DavidBanner
{
    public DavidBanner()
    {
        System.out.println("Mr. McGee, don't make me angry.");
    }

    public void speak()
    {
        System.out.println("You wouldn't like me when I'm angry.");
    }
}

public class IncredibleHulk extends DavidBanner
{
    public void speak()
    {
        System.out.println("Roar!");
    }
}
```

What is the output produced by the following statements?

```
IncredibleHulk hulk = new IncredibleHulk();
hulk.speak();
```

- (a) Mr. McGee, don't make me angry.
You wouldn't like me when I'm angry.
 - (b) Mr. McGee, don't make me angry.
Roar!
 - (c) You wouldn't like me when I'm angry.
 - (d) Roar!
11. Which class is a checked exception?
- (a) `ArrayIndexOutOfBoundsException`
 - (b) `IOException`
 - (c) `NullPointerException`
 - (d) `StringIndexOutOfBoundsException`

12. Which of the following is one of the three stream objects associated with devices that Java creates when a program begins executing?
- (a) `System.input`
 - (b) `System.output`
 - (c) `System.err`
 - (d) None of the above
13. What class from the `javax.swing` package is often extended to produce the top-level window of a GUI-based desktop application?
- (a) `Frame`
 - (b) `JFrame`
 - (c) `Panel`
 - (d) `TextField`

14. Given the following two constructors for the `Complex` class:

```
public Complex(double r, double i)
{
    this.real = r;
    this.imaginary = i;
}

public Complex()
{
    /* Insert line of code here */
}
```

Which line of code could be inserted into the no argument constructor to make it create a `Complex` object with the real part and imaginary part both equal to 0?

- (a) `this();`
 - (b) `this(0,0);`
 - (c) `super(0,0);`
 - (d) `return new Complex(0,0);`
15. What Java keyword is used in a class declaration to indicate that the class will be a subclass of another class?
- (a) `extends`
 - (b) `implements`
 - (c) `interfaces`
 - (d) `subclasses`

16. The `BasePlusCommissionEmployee` class is to be rewritten using an inheritance relationship rather than composition.

```
public class BasePlusCommissionEmployee
{
    private CommissionEmployee commissionEmployee;
    private double baseSalary;

    public BasePlusCommissionEmployee(String first, String last, String ssn,
        double sales, double rate, double salary)
    {
        commissionEmployee = new CommissionEmployee(first, last, ssn, sales, rate);
        baseSalary = salary;
    }
}

public class BasePlusCommissionEmployee extends CommissionEmployee
{
    private double baseSalary;

    public BasePlusCommissionEmployee(String first, String last, String ssn,
        double sales, double rate, double salary)
    {
        /* Insert line of code here */
        baseSalary = salary;
    }
}
```

Which line of code should be inserted to complete the rewritten six-argument constructor?

- (a) `this();`
 - (b) `this(first, last, ssn, sales, rate);`
 - (c) `super(first, last, ssn, sales, rate);`
 - (d) `return new CommissionEmployee(first, last, ssn, sales, rate);`
17. What type of relationship exists between the classes `Chocolate`, `PeanutButter`, and `ReesesCup` if the `ReesesCup` class has member variables of type `Chocolate` and `PeanutButter`?
- (a) Inheritance
 - (b) Composition
 - (c) Dependency
 - (d) Realization

18. Consider the following Java class:

```
1 public abstract class Art
2 {
3     public String name;
4     public double value;
5
6     public String toString()
7     {
8         return String.format("name=%s value=%s ",name,value);
9     }
10
11     public static void main(String args[])
12     {
13         Art pollock = new Art();
14         pollock.name = "No. 5, 1948";
15         pollock.value = 1.518E8;
16         System.out.println(pollock);
17     }
18 }
```

What is the output when the class is compiled and run?

- (a) name=No. 5, 1948 value=1.518E8
 - (b) Compilation error on line 6
 - (c) Compilation error on line 13
 - (d) An exception is thrown at runtime
19. What must be done to prevent classes which implement the following interface from modifying the values of the fields?

```
public interface PhysicalConstant
{
    double SPEED_OF_LIGHT = 2.99792458e8;
    double IDEAL_GAS_CONSTANT = 8.314472;
    double PLANCKS_CONSTANT = 6.62606896e-34;
    double AVOGADROS_NUMBER = 6.0221415e23;
}
```

- (a) Add the modifier **final** to each field declaration
- (b) Add the modifier **static** to each field declaration
- (c) Add the modifiers **final** and **static** to each field declaration
- (d) Nothing must be done

20. Consider the following Java class:

```
import java.io.*;

public class ExceptionCatcher
{
    public static void main(String args[])
    {
        String filename = "Foo.java";
        try
        {
            FileReader foo = new FileReader(filename);
        }
        catch(FileNotFoundException fnfe)
        {
            System.out.println("The file " + filename + " cannot be found.");
        }
        finally
        {
            System.out.println("An exception has occurred.");
        }
    }
}
```

If the file `Foo.java` does not exist, what is the output when the class is compiled and run?

- (a) The file `Foo.java` cannot be found.
 - (b) An exception has occurred.
 - (c) The file `Foo.java` cannot be found.
An exception has occurred.
 - (d) Compilation error
21. Which of the following has the items in the correct order for a valid Java source code file?
- (a) class declarations, package declaration, import declarations
 - (b) import declarations, package declaration, class declarations
 - (c) package declaration, class declarations, import declarations
 - (d) package declaration, import declarations, class declarations

22. Given the following two Java classes which are already compiled and in the classpath:

```
package car.japan;

public class Honda
{
    public static void printSlogan()
    {
        System.out.println("Honda: The Power of Dreams");
    }
}

package car.germany;

public class Volkswagen
{
    public static void printSlogan()
    {
        System.out.println("Volkswagen: Das Auto");
    }
}
```

What is the output when the following application is compiled and run?

```
1 import car.germany.*;
2
3 public class Automobile
4 {
5     public static void main(String args[])
6     {
7         car.japan.Honda.printSlogan();
8         Volkswagen.printSlogan();
9     }
10 }
```

- (a) Honda: The Power of Dreams
Volkswagen: Das Auto
- (b) Compilation error on line 1
- (c) Compilation error on line 7
- (d) An exception is thrown at runtime

23. Given the following Java class which is already compiled and in the classpath:

```
package cigar.cuba;

public class Cohiba
{
    public String toString()
    {
        return new String("Handmade from the finest tobacco available in Cuba");
    }
}
```

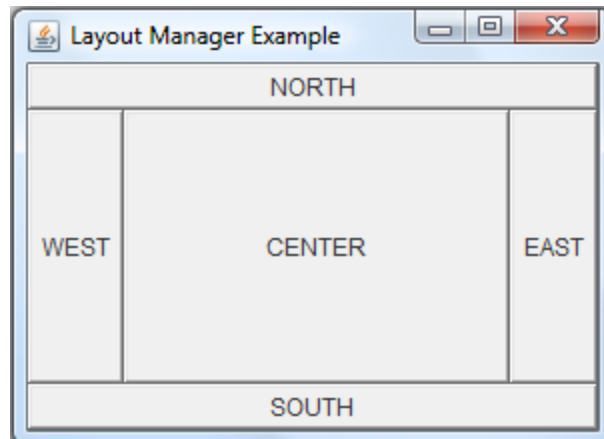
What must be done to make the following application compile and run?

```
import cigar.cuba.Cohiba;

public class CigarAficionado
{
    public static void main(String args[])
    {
        System.out.println(new Cohiba());
    }
}
```

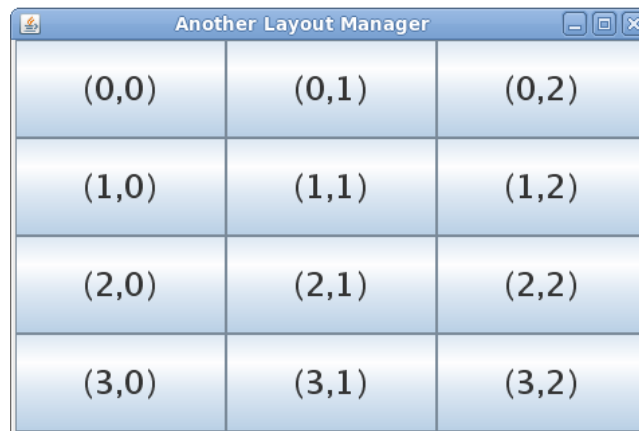
- (a) Nothing must be done to make the application compile and run
 - (b) Move the `CigarAficionado` class to the `cigar` package
 - (c) Make the `CigarAficionado` class a subclass of the `String` class
 - (d) Change the access modifier of the `Cohiba` class
24. What method can be implemented to allow for event-handling capabilities to be associated with a `JButton` ?
- (a) `ActionEvent`
 - (b) `ActionListener`
 - (c) `actionPerformed`
 - (d) `ActionHandler`
25. The process by which an entire object may be written to a file is called
- (a) Abstraction
 - (b) Encapsulation
 - (c) Realization
 - (d) Serialization

26. Which layout manager is depicted in the following figure?



- (a) BorderLayout
- (b) FlowLayout
- (c) GridLayout
- (d) DefaultLayout

27. Which layout manager is depicted in the following figure?



- (a) BorderLayout
- (b) FlowLayout
- (c) GridLayout
- (d) DefaultLayout

28. Consider the following Java class:

```
package animal.mammal.marine;

public class Whale
{
    private String getMantra()
    {
        return "Save the Whales";
    }
}
```

From which of the following can a method access the `getMantra` method

- (a) Inside the `Whale` class
 - (b) A subclass of the `Whale` class
 - (c) A class in the `animal.mammal.marine` package
 - (d) A class in the *default* package which doesn't extend the `Whale` class
29. What is the output when the following Java class is compiled and run?

```
public class FrayedKnot
{
    public static void main(String args[])
    {
        FrayedKnot f = new FrayedKnot();
        System.out.println(f);
        System.out.println("We don't serve Strings.");
    }

    public String toString()
    {
        return String.format("I'm afraid not!");
    }
}
```

- (a) I'm afraid not!
We don't serve Strings.
- (b) FrayedKnot@3e25a5
We don't serve Strings.
- (c) Compilation error
- (d) An exception is thrown at runtime

30. Given the following Java class:

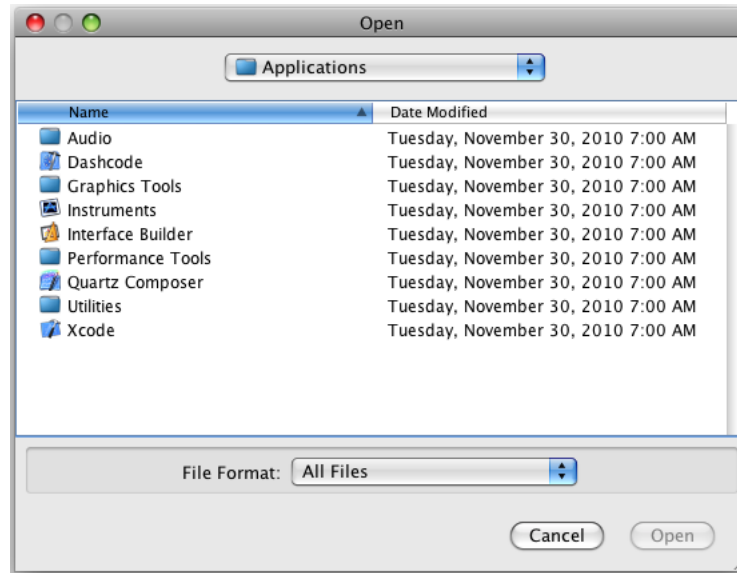
```
import java.io.*;
import java.util.*;

public class TextFileReader
{
    public static void main(String args[])
    {
        try
        {
            /* Insert line of code here */
            Scanner s = new Scanner(f);
            while(s.hasNext()) System.out.println(s.nextLine());
        }
        catch(IOException ioe)
        {
            ioe.printStackTrace();
        }
    }
}
```

Which line of code could be inserted in order to read input from the file `textfile.txt`?

- (a) `File f = new File("textfile.txt");`
 - (b) `FileInputStream f = new InputStream("textfile.txt");`
 - (c) `FileReader f = new Reader("textfile.txt");`
 - (d) None of the above
31. Which modifier can be used to cause a member variable to be ignored during the serialization process?
- (a) `absolute`
 - (b) `public`
 - (c) `skip`
 - (d) `transient`
32. What class can be used to send data to any text-based stream?
- (a) `Formatter`
 - (b) `Outputter`
 - (c) `Scanner`
 - (d) `Streamer`

33. What Swing component is shown in the following figure?



- (a) JFileFormatter
 - (b) JFileStreamer
 - (c) JFileChooser
 - (d) JFileScanner
34. What is the output when the following Java class is compiled and run?

```
import java.io.*;

public class IdentityVerifier
{
    public static void main(String args[])
    {
        String s = new Object();
        if (s instanceof Object) System.out.println("Strings are Objects");
        if (s instanceof Serializable)
            System.out.println("Strings are Serializable");
    }
}
```

- (a) Strings are Objects
Strings are Serializable
- (b) Strings are Serializable
- (c) Compilation error
- (d) An exception is thrown at runtime

35. What is the output when the following Java class is compiled and run?

```
import java.io.*;

public class CornFlakes implements Serializable
{
    public static void main(String args[])
    {
        CornFlakes firstBowl = new CornFlakes();
        CornFlakes secondBowl = firstBowl.cerealize();
        System.out.println(firstBowl != secondBowl);
    }

    public CornFlakes cerealize()
    {
        CornFlakes cereal = null;
        try
        {
            ObjectOutputStream output = new ObjectOutputStream(
                new FileOutputStream("breakfast.ser"));
            output.writeObject(this);
            ObjectInputStream input = new ObjectInputStream(
                new FileInputStream("breakfast.ser"));
            cereal = (CornFlakes) input.readObject();
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
        finally
        {
            return cereal;
        }
    }
}
```

- (a) true
- (b) false
- (c) Compilation error
- (d) An exception is thrown at runtime

36. Given the following Java classes which are already compiled and in the classpath:

```
public abstract class Dog
{
    public void speak()
    {
        System.out.println("Bark ");
    }
}

public class Achilles extends Dog
{
    public void speak()
    {
        System.out.print("RUFF ");
    }
}

public class Chloe extends Dog
{
    public void speak()
    {
        System.out.print("woof ");
    }
}
```

What is the output when the following application is compiled and run?

```
public class DogSpeak
{
    public static void main(String args[])
    {
        Dog pets[] = new Dog[2];
        pets[0] = new Achilles();
        pets[1] = new Chloe();
        for (Dog d : pets) d.speak();
    }
}
```

- (a) Bark Bark
- (b) RUFF woof
- (c) Compilation error
- (d) An exception is thrown at runtime

37. What is the output if one attempts to compile the following Java class files and run the Simulation application?

```
public class Table
{
    private int capacity;

    public void setCapacity(int c)
    {
        capacity = c;
    }

    public int getCapacity()
    {
        return capacity;
    }
}

public class Simulation
{
    private Table tables[];

    public Simulation()
    {
        tables = new Table[20];
        for (int i=0; i < 20; i++)
            tables[i] = new Table();
        System.out.println("Tables have been initialized");
    }

    public static void main(String args[])
    {
        Simulation s = new Simulation();
        System.out.println("Simulation with 20 tables started");
    }
}
```

- (a) Simulation with 20 tables started
- (b) Tables have been initialized
Simulation with 20 tables started
- (c) Compilation error
- (d) An exception is thrown at runtime

38. The following Java class does not compile and run:

```
public class FootballTeam
{
    private String city;
    private String quarterback;

    public FootballTeam(String city)
    {
        setCity(city);
    }

    public void setCity(String city)
    {
        this.city = city;
    }

    public void setQuarterback(String quarterback)
    {
        this.quarterback = quarterback;
    }

    /* Insert line of code here */

    public static void main(String args[])
    {
        FootballTeam steelers = new FootballTeam("Pittsburgh");
        steelers.setCity("Pittsburgh");
        steelers.setQuarterback("Ben Roethlisberger");

        FootballTeam ravens = new FootballTeam();
        ravens.setCity("Baltimore");
        ravens.setQuarterback("Joe Flacco");
    }
}
```

Which line of code could be inserted to make the class compile and run?

- (a) `public this() {}`
- (b) `public super() {}`
- (c) `public FootballTeam() {}`
- (d) `public void FootballTeam() {}`

39. What is the output when the following Java class is compiled and run?

```
public class DivideByZero
{
    public static void main(String args[])
    {
        System.out.println("100 divided by 0 = " + (100 / 0));
    }
}
```

- (a) 100 divided by 0 = 0
- (b) 100 divided by 0 = Infinity
- (c) Compilation error
- (d) An exception is thrown at runtime

40. Given the following two Java class files:

```
public class Exam
{
    public static void finish()
    {
        System.out.println("Congratulations! You have finished the exam.");
    }
}

public class FinalExam extends Exam
{
    public static void main(String args[])
    {
        /* Insert line of code here */
    }
}
```

Which line of code could be inserted to call the `finish` method?

- (a) `Exam.finish();`
- (b) `FinalExam.finish();`
- (c) `finish();`
- (d) All of the above